From zero to first test in your own LAVA laboratory

(in less than 45 minutes)

Paweł Wieczorek

February 21, 2017

Samsung R&D Institute Poland

Agenda

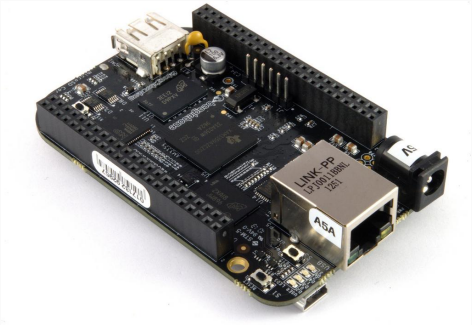
1. Introduction
2. Laboratory setup
3. Tools
4. Next steps
5. Conclusion

Introduction

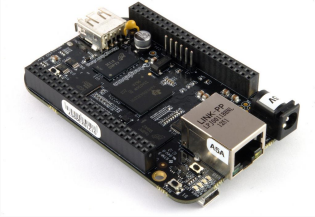
What is LAVA?

- Linaro Automated Validation Architecture
- Automation system for deploying operating systems
- Supports both physical and virtual hardware
- Allows running boot, bootloader and system level tests (extra hardware may be required)

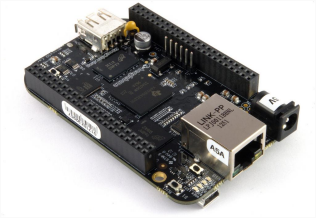
When is it needed? (Single target)



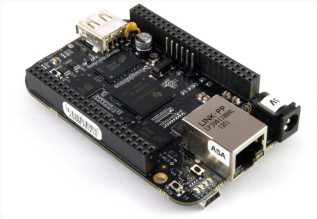
When is it needed? (Single instance)

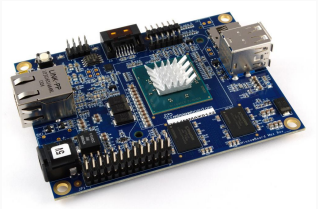


When is it needed? (Multiple targets)

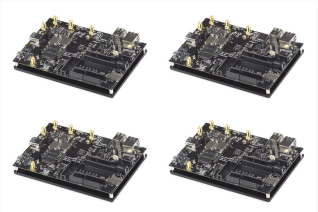


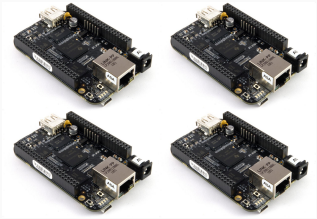
When is it needed? (New architecture)

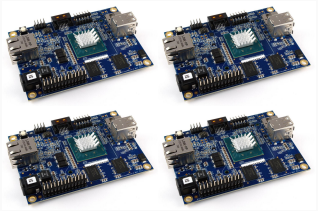


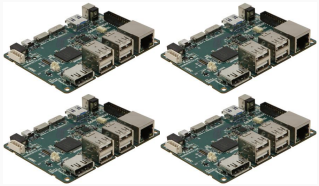


When is it needed? (Multiple instances)









When is it needed? (Abstraction layer)

What are LAVA use cases?

- Complex device management
- Resource allocation (various capabilities)
- Scheduling and dispatching tasks on numerous devices

How does it help?

- Provides unified device environment
- Allows test execution parallelization
- Collects and tracks results over time
- Supports direct device access
  - Hacking Sessions
  - Board Overseer

Who uses LAVA?

Linaro

Kernel CI

debian

Laboratory setup

Where to start?

- **Standalone** instance
- Virtual devices **only**
- Simple tests (**health checks**)

Rationale

- **Reduce** initial complexity
- **Familiarize** with the new workflow
- **Understand** LAVA concepts
- **Postpone** learning how to write tests

Requirements

Machine with supported Debian release (Ubuntu support frozen)

Distribution	Codename	Number	Support
Debian	experimental	n/a	Yes [1]
Debian	Sid (unstable)	n/a	Yes
Debian	Stretch (testing)	n/a	Yes [2]
Debian	Jessie (stable)	8.0	Yes [3]

Necessary files

- System image
- Health check job
- Device type template
- Device dictionary (instance definition):

```
{% extends 'qemu.jinja2' %}
{% set mac_addr = '52:54:00:12:34:AB' %}
{% set memory = '1024' %}
```

Installation – Host system

Step #1: Database & metapackage

```
# apt install postgresql
# apt install lava
```

Step #2: Enable access via web UI

```
# a2dissite 000-default.conf
# a2enmod proxy
# a2enmod proxy_http
# a2ensite lava-server.conf
# service apache2 restart
```

Configuration – Administrator

Step #3: Add main laboratory operator

```
# lava-server manage createsuperuser
```

Configuration – Devices

Step #4: Add devices to LAVA laboratory

```
# lava-server manage add-device-type qemu
# lava-server manage add-device
  --device-type qemu qemu01
# lava-server manage device-dictionary
  --hostname qemu01 --import qemu01.dict
```

Executing tests

CLI

```
$ lava-tool submit-job --help
```

Web UI

Tools

Configuration management

- Environment reproducibility
- Same on staging and production environment
- Choose personal favourite
- LAVA laboratory roles coming to Ansible Galaxy soon

Virtual machine management

- New machines brought up instantly
- Wide range of prebuilt boxes (careful!)

- Flexible (covers various use cases)
- User-friendly CLI/GUI tools

Next steps

Further details

- **Adding new device types to LAVA**
  `https://validation.linaro.org/static/docs/v2/first-devices.html#adding-new-device-types`
- **Writing tests**
  `https://validation.linaro.org/static/docs/v2/developing-tests.html`
- **Add your lab to Kernel CI**
  `https://github.com/kernelci/lava-ci#add-your-lab-to-kernelci`

Good reads

- **AGL Testframework setup instructions**
  `https://wiki.automotivelinux.org/agl-testframework/setup`
- **Civil Infrastructure Platform Testing initiative**
  `https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting`

Interesting talks

- Getting Started in LAVA V2 – Bill Fletcher (LAS16-TR05)
- Building a Boards Farm: Continuous Integration and Remote Control – Antoine Tenart & Quentin Schulz (ELCE 2016)
- Testing with volcanoes - Fuego+LAVA – Jan-Simon Möller (ELC 2017)

Help?

- **Comprehensive documentation**
  `https://validation.linaro.org/static/docs/v2`
- **Lava-users mailing list**
  `https://lists.linaro.org/mailman/listinfo/lava-users`
- **#linaro-lava on Freenode**
  `http://webchat.freenode.net/?channels=linaro-lava`

Conclusion

Summary

- Easy installation thanks to package repositories
- Instant setup (once all requirements are met)
- Environment unification for various device types
- No cost test execution parallelization
- Responsibilities division (farm maintained by its operators)

Final thoughts

- Exhaustive documentation has no downsides
- No need to reinvent the wheel in board farm management
- Automation always pays off in the long term

Questions?

Thank you!

Paweł Wieczorek

p.wieczorek2@samsung.com

Samsung R&D Institute Poland

Acknowledgements

- Bill Fletcher's YouTube channel
- Senthil Kumaran S – personal website
- Metropolis – simple, modern Beamer theme

Pictures used

- https://validation.linaro.org/static/docs/v2/_images/lava.svg
- https://c1.staticflickr.com/4/3845/14491195107_80cc27784a_b.jpg
- http://core0.staticworld.net/images/article/2016/04/3.artik-10-developer-board-100657852-large.jpg
- https://upload.wikimedia.org/wikipedia/commons/e/ee/Hardkernel_Odroid_XU3_Board.jpg
- https://c2.staticflickr.com/4/3862/14903812841_575f03a6c4_b.jpg
- https://www.linaro.org/wp-content/uploads/2014/03/RGB-Linaro_Standard.png
- https://upload.wikimedia.org/wikipedia/commons/4/4a/Debian-OpenLogo.svg
- https://wiki.automotivelinux.org/_media/wiki/logo.png
- https://upload.wikimedia.org/wikipedia/commons/4/45/Qemu_logo.svg
- https://upload.wikimedia.org/wikipedia/commons/0/05/Ansible_Logo.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/8/87/Vagrant.png/394px-Vagrant.png
- https://upload.wikimedia.org/wikipedia/commons/thumb/d/da/Libvirt_logo.svg/800px-Libvirt_logo.svg.png